

Rapid Loop Updates

Vadim Indelman and Frank Dellaert

Technical Report GT-RIM-CP&R-2012-001, September 11, 2012

1 Introduction

An important component of any simultaneous localization and mapping (SLAM) system, as well as modern vision-aided navigation systems, is the ability to efficiently incorporate loop-closure measurements. EKF-SLAM and full-SLAM are, arguably, the two common approaches that have been developed over the years. The former maintains the observed landmarks and the current pose while marginalizing out past robot poses. The latter, maintains both the observed landmarks and all robot's poses - past poses and current pose.

Fusing loop-closure measurements both in EKF-SLAM and full-SLAM typically involves considerable computations causing to non-negligible delays, depending on the number of involved variables. Since many robotic applications require a navigation solution at a high frequency, it was proposed [7, 6] to incorporate loop closure measurements in a background process while fusing any other measurements in a foreground high-rate process. Thus, in [7] an EKF is used for processing incoming IMU and visual measurements, and bundle adjustment (BA) is applied in a background process whenever a loop-closure measurement is obtained, or from time to time for limiting linearization errors. More recently, the authors of [6] show how these two processes (filtering and smoothing) can be parallelized into a single optimization problem, which was not addressed previously.

Incorporating loop-closure measurements can thus be performed while still operating at high-frequency [6]. However, the navigation solution calculated by the filter is not updated with loop-closure information until the smoother has finished processing the loop-closure measurement in a background process. Thus, during that time the solution calculated by the filter does not utilize all the available measurements and thus is sub-optimal.

Rapid loop updates (RLU) is a suggested approach to address this issue. Upon obtaining a loop closure measurement, the idea is to update the filter with an approximate loop-closure update, while the smoother is engaged in calculating the exact update. Calculating this approximate update is expected to be computationally cheap and fast, since it only involves variables that are related by the loop-closure measurement. This is in contrast to calculating the exact update by the smoother, which typically requires optimizing all the

variables in the loop chain. Once the smoother finishes calculating the exact loop-closure update, this update can be consistently incorporated into the filter by first canceling the approximate RLU update.

This technical report introduces the concept of RLU and discusses related future research directions.

2 Problem Formulation

Denote the robot's state vector at time t_k by x_k and let X_k represent all the robot's state vectors up to time t_k :

$$X_k \doteq \begin{bmatrix} x_1^T & \dots & x_k^T \end{bmatrix}^T.$$

Let $p(X_k|Z_k)$ denote the joint probability function of X_k given all the measurements $Z_k \doteq \{z_i\}_{i=1}^k$ up to time t_k . The maximum a posteriori (MAP) estimate of X_k is then given by

$$\hat{X}_k = \arg \max_{X_k} p(X_k|Z_k).$$

Each new measurement z_i can be related to several variables in X_k . Denoting these variables by X_{z_i} , a general measurement model can be written as

$$z_i = h_i(X_{z_i}) + v_i,$$

with v_i representing a measurement noise with a covariance matrix Σ_i . Incorporating a new measurement into the optimization and calculating an updated MAP estimate can be done efficiently using incremental smoothing [5]. Typically only a small part of the variables are updated.

However, as described in the introduction, when the new measurement is a loop-closure measurement, the number of variables involved in the incremental optimization increases may involve non-negligible computational time, during which the current navigation solution is not incorporated with any loop-closure information.

In the next section we describe a method to calculate an approximate loop-closure update for the current navigation state, that can be used while the exact update is being calculated by the incremental optimization (smoother).

3 Rapid Loop Updates

3.1 Calculating RLU and Incorporating in the Filter

Consider the following general model for a loop-closure measurement

$$z_L = h_L(X_L) + v \tag{1}$$

that can be represented by the factor

$$f_L(X) = \exp(d(h(X_L) - z))$$

where $X_L \subset X$ are the variables involved in the loop-closure measurement model, and $d(a)$ denotes a certain cost function. Assuming a Gaussian distribution, this cost function is the squared Mahalanobis distance $d(a) \doteq a^T \Sigma^{-1} a$, with Σ being the measurement covariance matrix. For example, a robot can re-observe a landmark, in which case X_L will comprise the observing camera pose and the observed landmark: $X_L \doteq \{x, l\}$

Let $X_L^{RLU} \subset X_L$ denote the variables to be updated by RLU and $\bar{X}_L^{RLU} \doteq X_L \setminus X_L^{RLU} \subset X_L$ be the rest of the variables in X_L . In the previous example, X_L^{RLU} can be the camera pose, in which case \bar{X}_L^{RLU} will be the landmark.

Denote by $\hat{X}_L^{RLU}, \hat{\bar{X}}_L^{RLU}$ the current linearization points of $X_L^{RLU}, \bar{X}_L^{RLU}$.

RLU is an approximate update, that is calculated only for X_L^{RLU} and making use only of the loop-closure factor f_L . Assuming a Gaussian distribution, our goal is to find Δ , the delta in \hat{X}_L^{RLU} , such that

$$\left\| h_L \left(\hat{X}_L^{RLU} + \Delta, \hat{\bar{X}}_L^{RLU} \right) - z_L \right\|_{\Sigma}^2 \approx \left\| h_L \left(\hat{X}_L^{RLU}, \hat{\bar{X}}_L^{RLU} \right) - z_L + A_L \Delta_L^{RLU} \right\|_{\Sigma}^2 \quad (2)$$

is minimized. Here A_L is the Jacobian matrix with respect to X_L^{RLU} , evaluated about the linearization point $\hat{X}_L^{RLU}, \hat{\bar{X}}_L^{RLU}$.

Eq. (2) can be considered as a unary linear factor on the variables X_L^{RLU} :

$$f_{RLU}(X_L^{RLU}) \doteq \exp \left(\| A_L \Delta_L^{RLU} - \check{z}_L \|_{\Sigma}^2 \right),$$

with $\check{z}_L \doteq z_L - h_L \left(\hat{X}_L^{RLU}, \hat{\bar{X}}_L^{RLU} \right)$, and can therefore be directly incorporated in the inference procedure. Alternatively, Δ_L^{RLU} can be calculated directly as the standard least-squares solution

$$\hat{\Delta}_L^{RLU} = (A_L^T A_L)^{-1} b_L$$

with the right hand side (rhs) vector $b_L \doteq A_L^T \check{z}_L$, and used for updating \hat{X}_L^{RLU} :

$$\hat{X}_L^{RLU} \leftarrow \hat{X}_L^{RLU} + \hat{\Delta}_L^{RLU}.$$

3.2 When RLU is a Good Approximation?

The approximate update calculated by RLU is expected to be close to the exact update of the smoother, when the uncertainty associated with the state X_L^{RLU} is considerably higher than the uncertainty of the other states (\bar{X}_L^{RLU}) that are involved in the loop-closure measurement.

3.3 Incorporating an Exact Update and Canceling RLU

Once the smoother finishes processing the loop-closure measurement, it is possible to incorporate an exact loop-closure update into the filter [7, 6].

First, however, the approximate update (RLU) should be cancelled. Consider all the factors in the sliding window of the filter at the time the RLU update should be canceled and denote the Jacobian matrices and the rhs vectors of each of these factors by A_i and b_i . Assume the number of such factors, including the RLU update, is n . Let also X_f be the filter's state vector, comprising states from the mentioned sliding window. The current value of X_f was calculated using all the available n measurements. Assuming no linearization is done, the delta vector Δ that was used to update X_f to its current value is given by:

$$H\Delta = b$$

with

$$H \doteq \sum_i A_i^T A_i \quad , \quad b \doteq \sum_i b_i.$$

Canceling the RLU update, assuming the Hessian matrix $A_L^T A_L$ and rhs vector b_L of the RLU update are stored, is equivalent to updating the matrix H and the vector b according to

$$H \leftarrow H - A_L^T A_L \quad , \quad b \leftarrow b - b_L.$$

This procedure can be accordingly adapted in case the square root information matrix R ($A = R^T R$) instead of the Jacobian matrix A is maintained.

Using a factor graph formulation, the RLU update may be considered as a unary factor, while the above procedure of canceling an RLU update can be realized using an anti-factor. This anti-factor can be easily calculated from the original RLU factor.

3.4 Incorporating RLU within Concurrent Smoothing and Filtering

This section provides a high-level overview of the proposed approach for incorporating the RLU concept within the recently developed approach for concurrent smoothing and filtering [6]. In [6], the smoother and the filter are run in parallel and solve a single optimization problem. The problem is formulated in terms of a factor graph, and solved using incremental smoothing [5]. The optimization problem is represented by a single Bayes tree [4] and the parallelization of smoothing and filtering is obtained by identifying a group of variables that separate the full joint probability function $p(X)$ into two groups:

$$p(X) = p(X_R|X_S) p(X_S) p(X_f|X_S) \quad (3)$$

where X_f are the state vectors of the filter, X_S are the separator variables, and X_R are the rest of the variables that are solved by the smoother. The complete set of variables is denoted by $X = \{X_R, X_S, X_f\}$. The Bayes tree representation equivalent to the factorization (3) is given in Figure 1a.

Figure 1b illustrates a loop closure measurement between some variables $X'_f \subset X_f$ in the filter and some variables $X_n \subset X_R$ in the smoother:

$$z_L = h_L(X'_f, X_n) + v$$

. The equivalent loop-closure factor is $f_L(X'_f, X_n)$. Using the notations from Section 3.1, the variables $X_L \subset X$ that are involved in the loop-closure measurement model are $X_L = \{X'_f, X_n\}$, the variables to be updated by RLU are $X_L^{RLU} = X'_f$, while the rest of the variables in X_L are $\bar{X}_L^{RLU} = X_n$.

Applying RLU in this example is equivalent to adding a unary factor $f_{RLU}(X'_f)$ to the filter's branch of the Bayes tree, as illustrated in Figure 1c.

The story is far from being complete, since the exact update from the smoother while canceling the RLU update should still be incorporated into the filter. It is our belief that this process can be realized by accordingly modifying the synchronization procedure between the smoother and the filter [6], which is the subject of ongoing research.

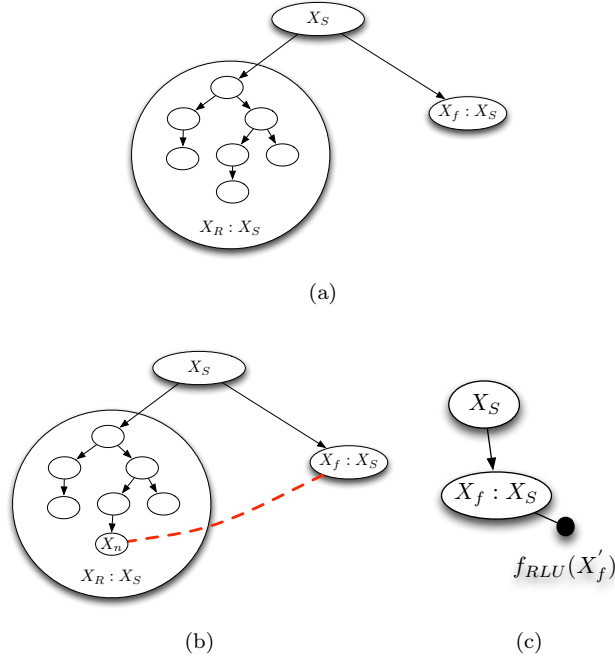


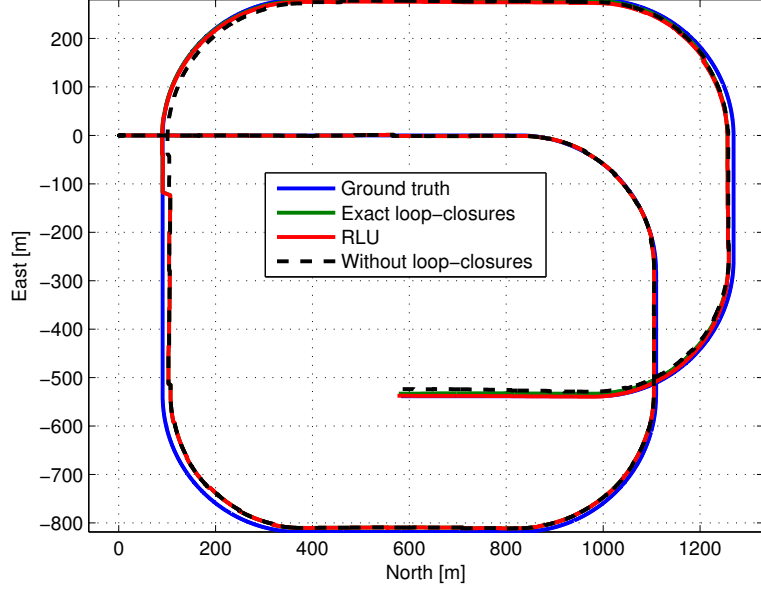
Figure 1: (a) Bayes tree representation for concurrent smoothing and mapping. (b) Loop closure measurement between some state in the filter and some state X_n in the smoother. (c) Adding RLU unary factor to the filter branch of the Bayes tree.

4 Preliminary Results

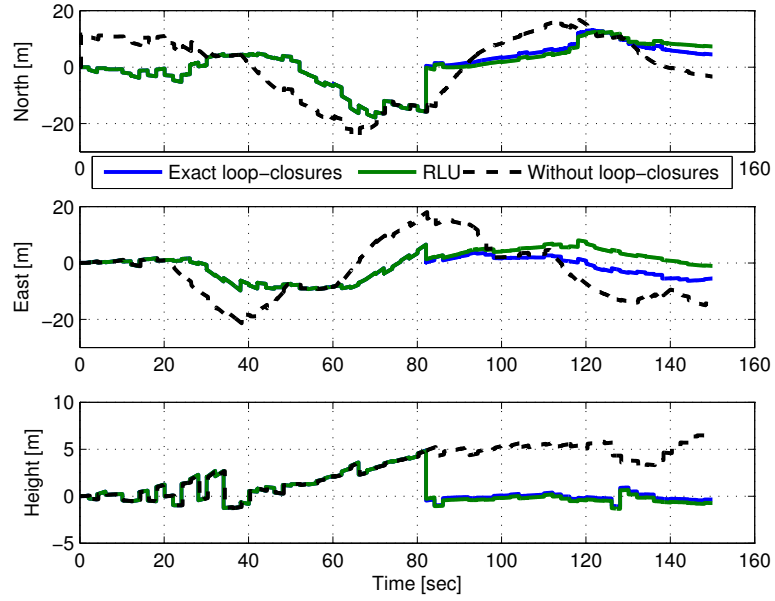
A preliminary concept of RLU was developed in [1, 2] in the context of vision-aided navigation based on multi-view constraints. Visual observations of unknown landmarks were represented by three-view constraints between the appropriate views. These constraints were used for updating the current navigation solution without applying smoothing for calculating the exact solution. Specifically, given a loop-closure measurement that relates three overlapping views, the navigation states of the first two views were kept fixed, and the navigation state of the third view (representing the current navigation state) was updated using an RLU. The involved uncertainties in the past two states were taken into account during that procedure. The reader is referred to [1, 2] for further discussion and results.

In order to get some initial insights, a simplified version of RLU was implemented and examined in a factor graph formulation. An aerial scenario with a loop-pattern trajectory at a constant height was assumed (see Figure 2a), similarly to [6, 3]. The available sensors were: IMU and stereo camera. The latter produced relative-pose measurements at 2 Hz, while IMU measurements were incorporated at 100 Hz. Relative-pose measurements were also used as loop-closure measurements whenever previously-seen ground areas were re-observed. Figure 2 shows the results and provides a comparison between incorporating loop-closure information using incremental smoothing ('Exact loop-closures') and using RLU. As seen from Figure 2b, the estimation errors are of the same order in both cases.

Although these results are encouraging, an extensive performance evaluation is required for being able to draw any substantial conclusions.



(a)



(b)

Figure 2: (a) Ground truth and estimated trajectory (top view); (b) Position estimation errors.

References

- [1] V. Indelman. *Navigation Performance Enhancement Using Online Mosaicking*. PhD thesis, Technion - Israel Institute of Technology, 2011.
- [2] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein. Real-time vision-aided localization and navigation based on three-view geometry. *IEEE Trans. Aerosp. Electron. Syst.*, 48(3):2239–2259, July 2012.
- [3] V. Indelman, S. Williams, M. Kaess, and F. Dellaert. Factor graph based incremental smoothing in inertial navigation systems. In *Intl. Conf. on Information Fusion, FUSION*, 2012.
- [4] M. Kaess, V. Ila, R. Roberts, and F. Dellaert. The Bayes tree: An algorithmic foundation for probabilistic robot mapping. In *Intl. Workshop on the Algorithmic Foundations of Robotics*, Dec 2010.
- [5] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Intl. J. of Robotics Research*, 31:217–236, Feb 2012.
- [6] M. Kaess, S. Williams, V. Indelman, R. Roberts, J.J. Leonard, and F. Dellaert. Concurrent filtering and smoothing. In *Intl. Conf. on Information Fusion, FUSION*, 2012.
- [7] A.I. Mourikis and S.I. Roumeliotis. A dual-layer estimator architecture for long-term localization. In *Proc. of the Workshop on Visual Localization for Mobile Platforms at CVPR*, Anchorage, Alaska, June 2008.